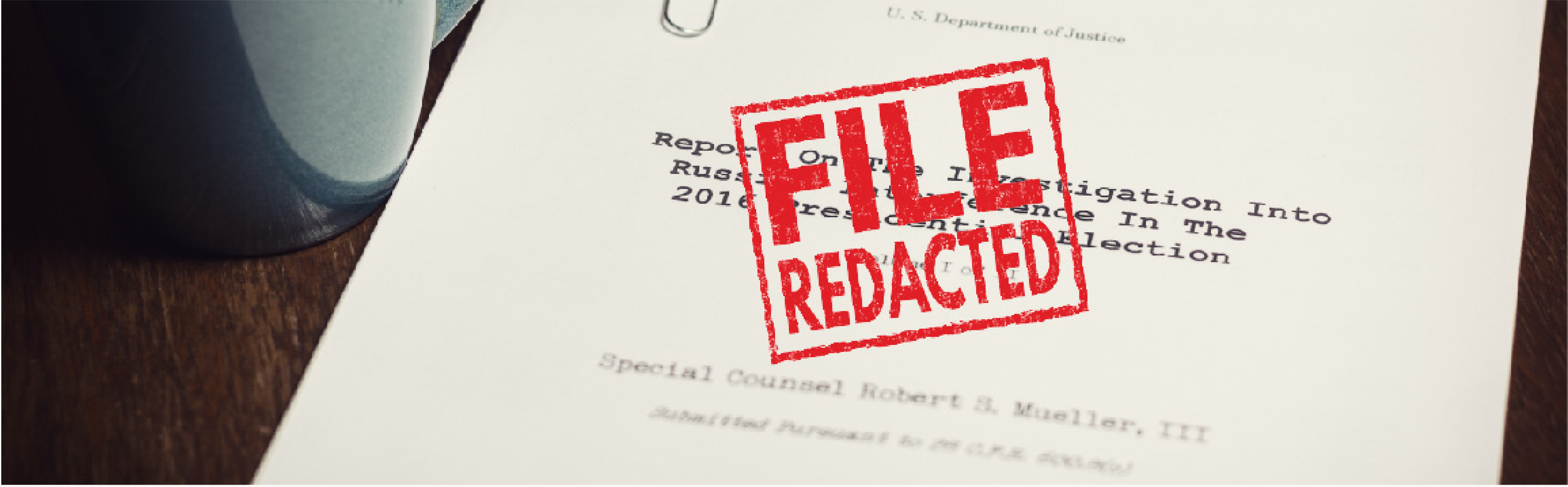

U. S. Department of Justice
FILE REDACTED
Special Counsel Robert S. Mueller, III

OctoberPDFest
ONLINE

High-Security PDF Redactions

Closing known routes for leakage in redacted documents

History of document redaction

OctoberPDFest ONLINE

- Early use of redact meant to organize or edit.
- Its usage as we understand it today is relatively new.
- From the Oxford English Dictionary:
- c. transitive. To censor (a document) by removing or blacking out certain words or passages prior to publication or release, esp. for legal, security, or confidentiality purposes; to remove or black out (words or information) in this way. Frequently in passive. Now the most common sense.
- 1958 – N.Y. Suppl. 2nd Ser. 168 422 (note) [We] agree that feasible means should have been adopted to redact DeGennaro's confession and admissions so as to restrict their contents to his own inculpations, and thus have avoided any possible prejudice to Lombard.

PDF association

Artifex

History of document redaction

- 1973 – One of the first recorded discussions to remove names from a testimony with blacked out names was discussed in a U.S. Court of Appeals case.
- 1988 – One of the first highly public use of redaction was Oliver North's notebooks during the Iran-Contra hearings.
- 1993 – Adobe introduces PDF.
- 2006 – Adobe published a technical note providing guidance on PDF redaction which involved black rectangles to cover regions, saving pages as images, and reassembling images into PDF pages.
- 2006 – PDF redaction annotation introduced as an addendum to version 1.7.
- 2019 – Fast forward to recent times and we have one of the most public uses of redaction and PDF with the Mueller report.

Many options for PDF redaction

OctoberPDFest ONLINE

soda PDF ANYWHERE
PDFTRON
ActivePDF®
PDFCOMPLETE
Smallpdf
ITEXT
nitro
ideagen
opentext™
DocsCorp Work smart
PSPDFKit
foxit®
iSkysoft
Artifex

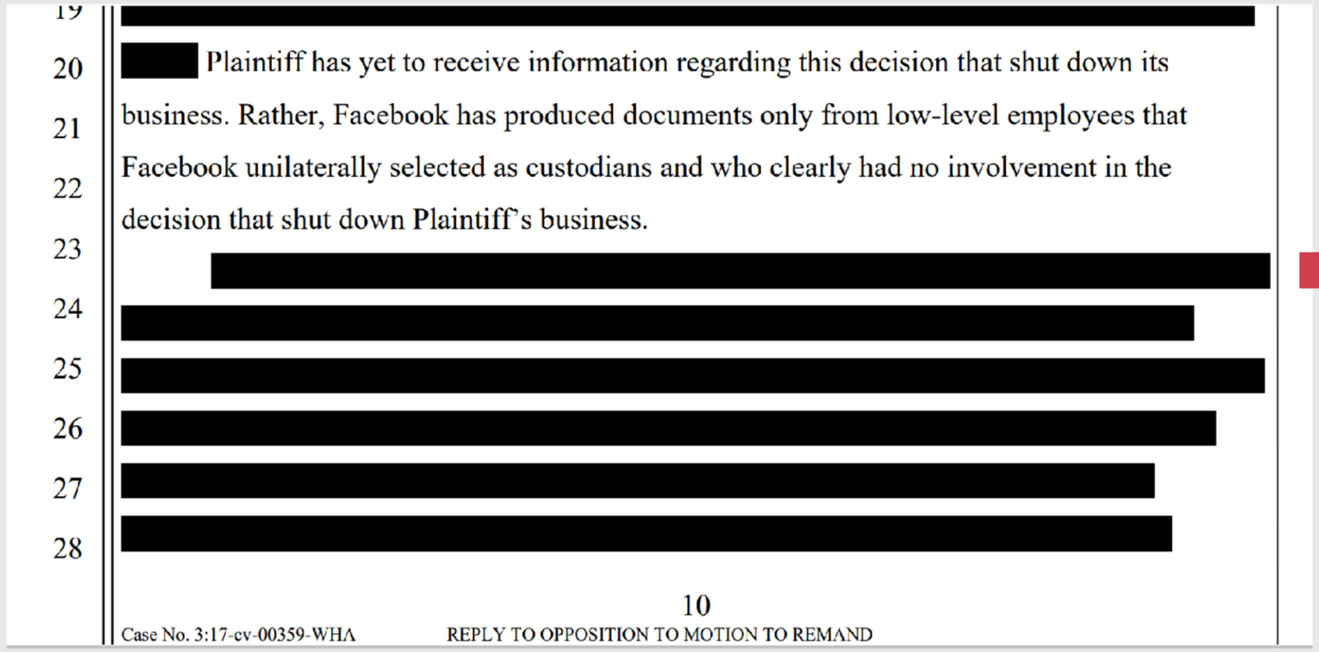
History of Failures

- 2009 – HSBC Exposed Sensitive Bankruptcy Data.
- 2009 – Associated Press was able to read redacted court testimony that showed Facebook's estimate of its own market value. The AP got its scoop by cutting and pasting the blacked-out sections of a court transcript.
- 2009 – TSA used black rectangles to cover text in a publicly available PDF that described screening methods designed to thwart terrorists.
- 2011 – British Ministry of Defense, published a PDF document online, unintentionally revealing information about nuclear submarine security. Added black background to "removed text sections".
- 2011 – A California federal district judge filed an opinion in a patent infringement suit by Apple against Samsung Electronics. Discussion of studies by Apple and details on Apple's licensing deals were supposedly redacted.
- 2014 – As part of its latest reporting on leaked NSA documents regarding Snowden, the New York Times did not properly redact a PDF that listed the name of a National Security Agency employee who prepared the document as well as a target of the program.
- 2018 – Improperly redacted PDF legal documents show Facebook considered selling deep access to data to other corporations.
- 2019 – Paul Manafort's lawyers submitted a PDF that had black boxes drawn over the text to be redacted but without deleting the actual text underneath it.

2018 Facebook example still available!

OctoberPDFest ONLINE

https://www.documentcloud.org/documents/5317193-7b6c2e96-396f-4cf6-ba18-65c8f848383d.html


[redacted]
[redacted] Plaintiff has yet to receive information regarding this decision that shut down its business. Rather, Facebook has produced documents only from low-level employees that Facebook unilaterally selected as custodians and who clearly had no involvement in the decision that shut down Plaintiff's business.

[redacted]

10

Case No. 3:17-cv-00359-WHA REPLY TO OPPOSITION TO MOTION TO REMAND

Nonetheless, Plaintiff has found in the files of these low-level employees clear evidence that beginning at least by 2012 and likely earlier, Facebook engaged in a scheme to trade access to data in its operating system on advantageous terms in order to extract large payments from collusive partners who sought to stifle competition in the various software markets in which they operated. Plaintiff has found direct evidence of agreements in which large companies agreed to spend substantial sums of money on advertising with Facebook in…

The difficulties

- Human error with use of software.
- Software error due to bugs.
- Easy for information to hide. PDF is a complex format.
- We don’t know what we don’t know.

Cost of failures can now be significant

- On 25 May 2018, the European Union’s General Data Protection Regulation (GDPR) became law.
- The regulation affects all member states of the EU and obliges them to be more accountable, transparent and responsible for the data they hold, process and share.
- Fines: Maxing at 20 million Euros or 4% of worldwide revenue from the preceding year, whichever is higher.

Many options for PDF redaction

OctoberPDFest ONLINE

- How do you know the redaction solution works?
- Do you know what the code is doing?

soda PDF ANYWHERE
PDFTRON
ActivePDF®
PDFCOMPLETE
Smallpdf
ITEXT
nitro
ideagen
opentext™
DocsCorp Work smart
PSPDFKit
foxit®
iSkysoft
Artifex

PDF redaction annotation

The redaction annotation involves two steps.

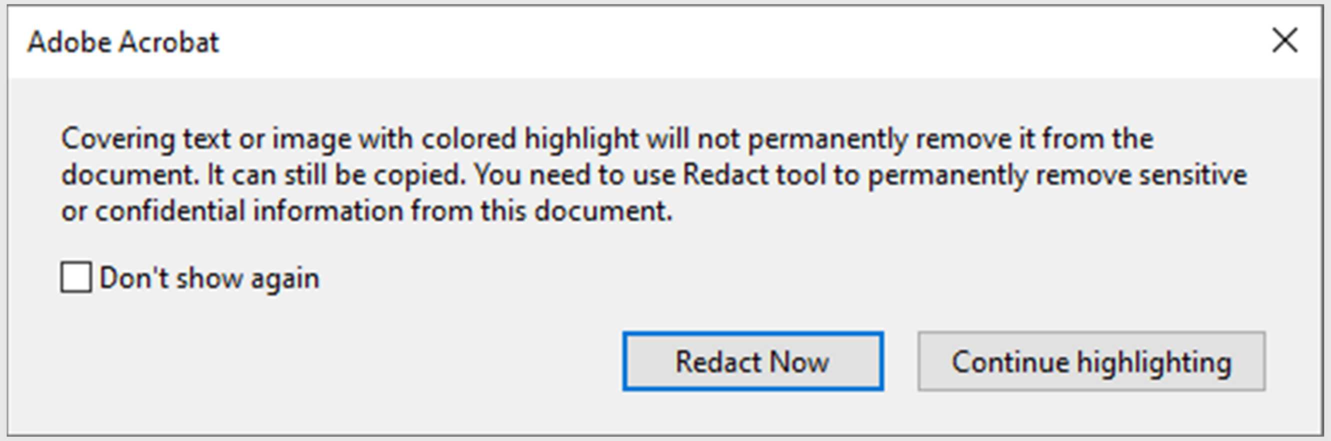
1) The region to be redacted is identified with a black rectangle.

Once the regions have been verified as being those of interest, the application must remove the information beneath the rectangle.

2) A “sanitize” step is used to remove meta-data.

Obvious failures

- Text contained underneath a solid rectangle.
- Most common failure of the past.
- If you do this today with Acrobat Pro you get message.

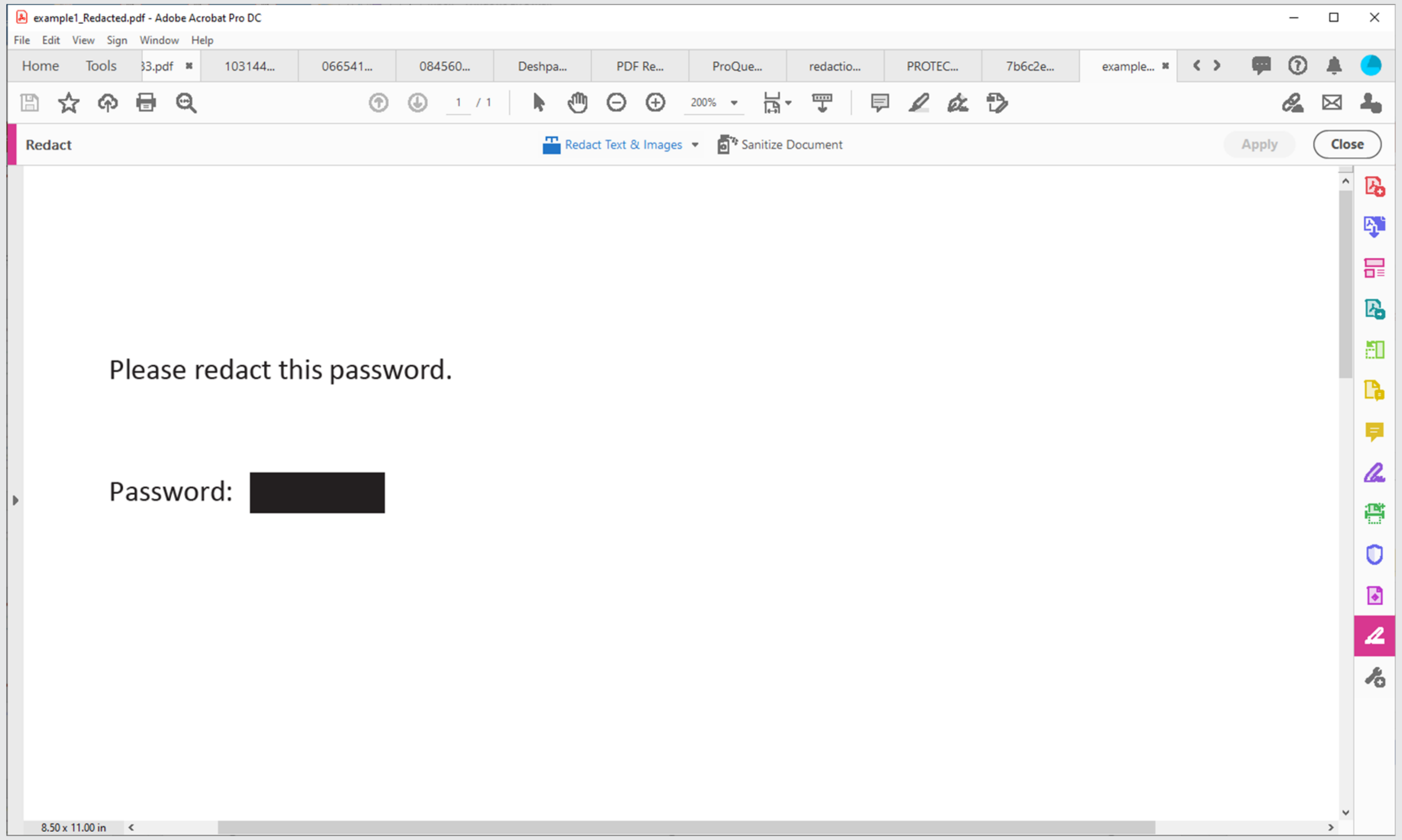


Covering text or image with colored highlight will not permanently remove it from the document. It can still be copied. You need to use Redact tool to permanently remove sensitive or confidential information from this document.

Less obvious failures

OctoberPDFest ONLINE

- Invisible text can be present



Please redact this password.

Password: [redacted]

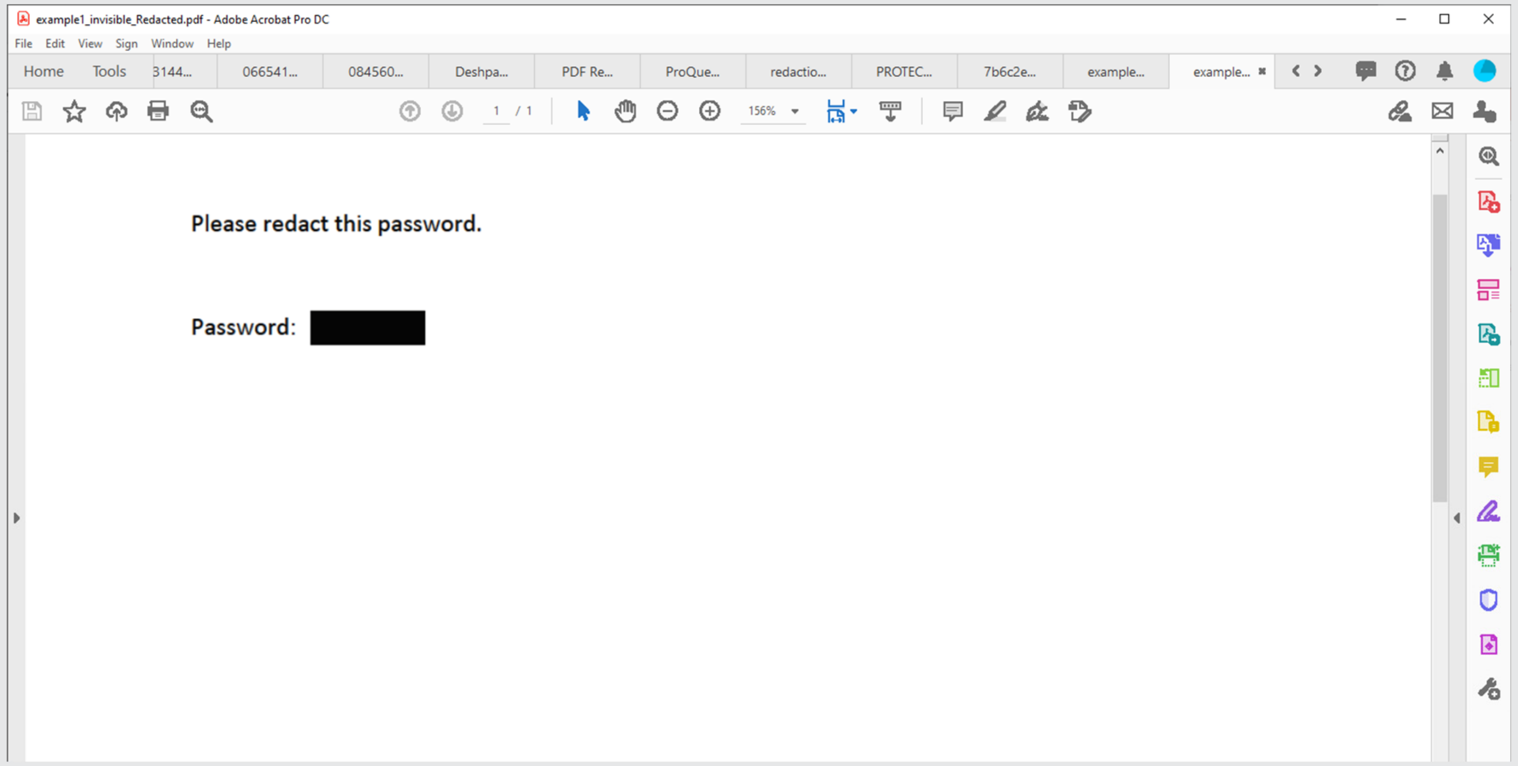
With no white text or transparent text present in file, page retains vector text.

PDF association

Artifex

Less obvious failures

- Invisible text can be present


example1_invisible_Redacted.pdf - Adobe Acrobat Pro DC
File Edit View Sign Window Help
Home Tools 3144... 066541... 084560... Deshpa... PDF Re... ProQue... redactio... PROTEC... 7b6c2e... example... example...
1 / 1
156%

Please redact this password.

Password: [redacted]

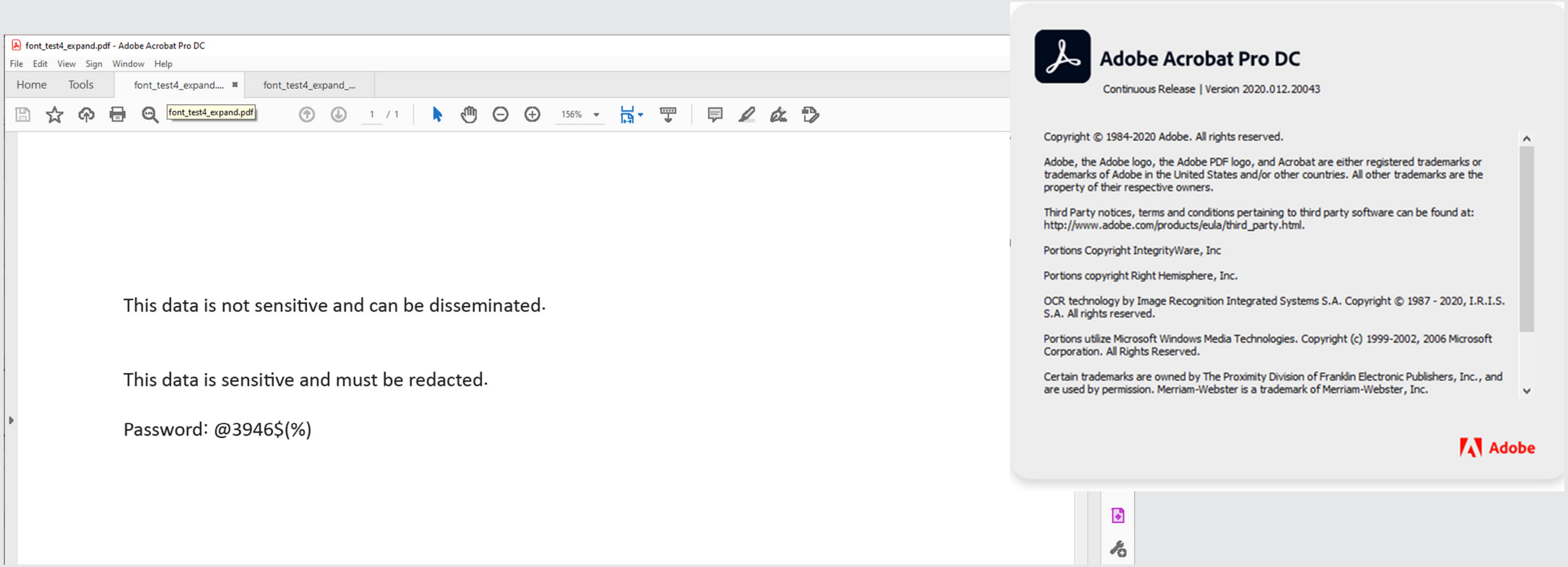
With white text, or transparent text present in file, redacted document is converted to image, if you select to sanitize and remove hidden information.

If you choose not to sanitize and remove hidden information, then that content is retained.

Even less obvious failures

OctoberPDFest ONLINE

- Subset fonts can retain information


font_test4_expand.pdf - Adobe Acrobat Pro DC
File Edit View Sign Window Help
Home Tools font_test4_expand.... font_test4_expand_...
font_test4_expand.pdf
156%

This data is not sensitive and can be disseminated.

This data is sensitive and must be redacted.

Password: @3946$(%)

Adobe Acrobat Pro DC

Continuous Release | Version 2020.012.20043

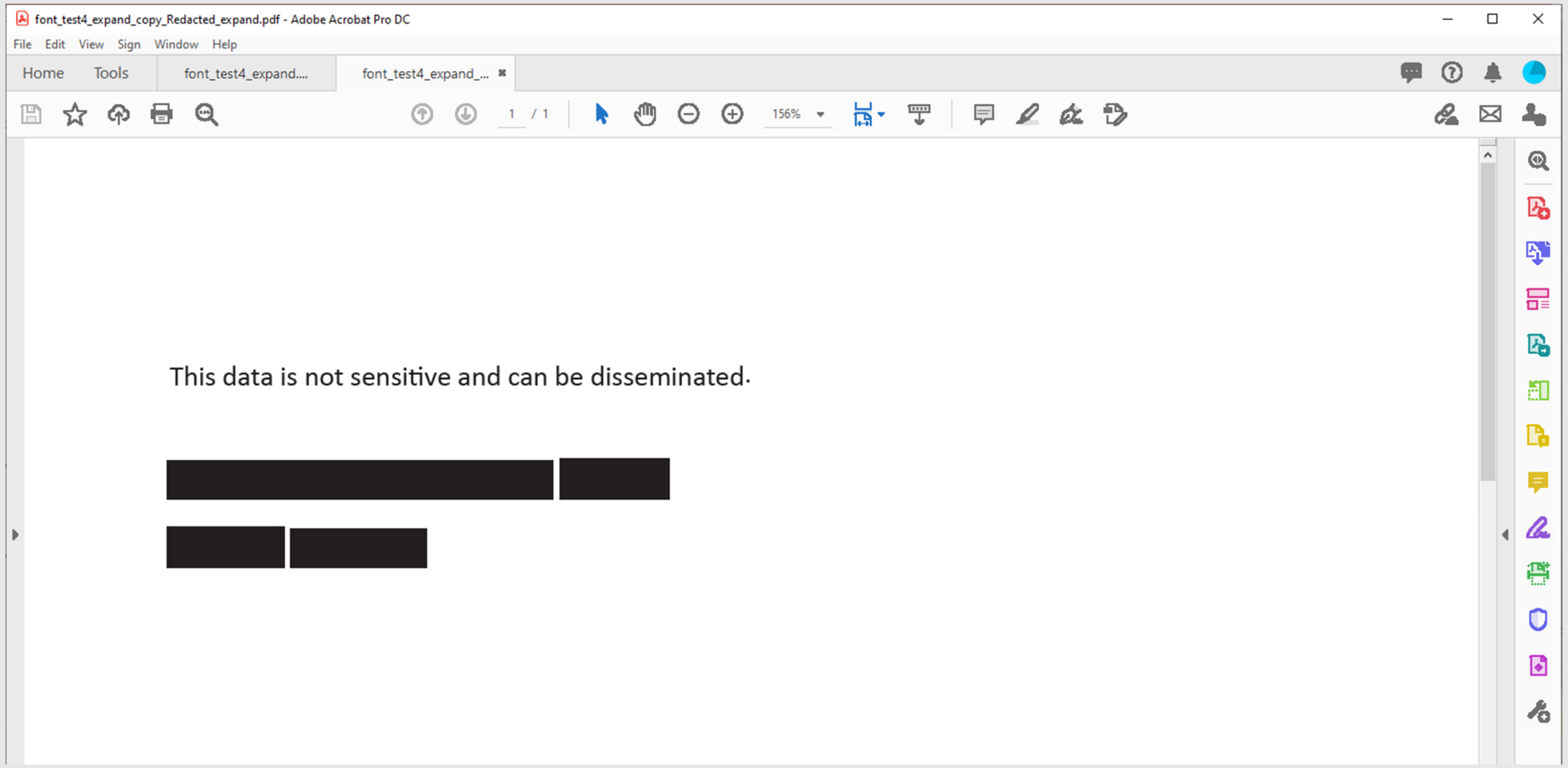
Adobe

PDF association

Artifex

Even less obvious failures

- Subset fonts can retain information


font_test4_expand_copy_Redacted_expand.pdf - Adobe Acrobat Pro DC
File Edit View Sign Window Help
Home Tools font_test4_expand.... font_test4_expand_...
1 / 1
156%

This data is not sensitive and can be disseminated.

[redacted]

[redacted]

Even less obvious failures

Close look at Unicode CMap in the redacted file.

```
15 0 obj
<</Length 759>>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CIDSystemInfo
<< /Registry (Adobe)
/Ordering (UCS) /Supplement 0 >> def
/CMapName /Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
31 beginbfchar
<0003> <0020>
<0102> <0061>
<010F> <0062>
<0110> <0063>
<011A> <0064>
<011E> <0065>
<0057> <0050>
<0064> <0054>
<015A> <0068>
<015D> <0069>
<0175> <006D>
<0176> <006E>
<017D> <006F>
<018C> <0072>
<0190> <0073>
<0357> <003A>
<0358> <002E>
<0439> <0025>
<019A> <0074>
<019F> <FFFD>
<01B5> <0075>
<037E> <0028>
<037F> <0029>
<01C0> <0076>
<01C1> <0077>
<039B> <0040>
<03A8> <0024>
<03EF> <0033>
<03F0> <0034>
<03F2> <0036>
<03F5> <0039>
endbfchar
```

Even less obvious failures

OctoberPDFest ONLINE

Close look at Unicode CMap in the redacted file.

```
15 0 obj
<</Length 759>>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CIDSystemInfo
<< /Registry (Adobe)
/Ordering (UCS) /Supplement 0 >> def
/CMapName /Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
```

```
31 beginbfchar
<0003> <0020>   Space
<0102> <0061>   a
<010F> <0062>   b
<0110> <0063>   c
<011A> <0064>   d
<011E> <0065>   e
<0057> <0050>   P
<0064> <0054>   T
<015A> <0068>   h
<015D> <0069>   i
<0175> <006D>   m
<0176> <006E>   n
<017D> <006F>   o
```

```
<018C> <0072>   r
<0190> <0073>   s
<0357> <003A>   :
<0358> <002E>   .
<0439> <0025>   %
<019A> <0074>   t
<019F> <FFFD>
<01B5> <0075>   u
<037E> <0028>   (
<037F> <0029>   )
<01C0> <0076>   v
<01C1> <0077>   w
<039B> <0040>   @
<03A8> <0024>   $
<03EF> <0033>   3
<03F0> <0034>   4
<03F2> <0036>   6
<03F5> <0039>   9
endbfchar
```

PDF association

Artifex

Even less obvious failures

OctoberPDFest ONLINE

Red characters retained in CMap but do not appear in document.

```
15 0 obj
<</Length 759>>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CIDSystemInfo
<< /Registry (Adobe)
/Ordering (UCS) /Supplement 0 >> def
/CMapName /Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
```

```
31 beginbfchar
<0003> <0020>      Space
<0102> <0061>      a
<010F> <0062>      b
<0110> <0063>      c
<011A> <0064>      d
<011E> <0065>      e
<0057> <0050>      P
<0064> <0054>      T
<015A> <0068>      h
<015D ><0069>      i
<0175> <006D>      m
<0176> <006E>      n
<017D> <006F>      o
```

```
<018C> <0072>      r
<0190> <0073>      s
<0357> <003A>      :
<0358> <002E>      .
<0439> <0025>      %
<019A> <0074>      t
<019F> <FFFD>
<01B5> <0075>      u
<037E> <0028>      (
<037F> <0029>      )
<01C0> <0076>      v
<01C1> <0077>      w
<039B> <0040>      @
<03A8> <0024>      $
<03EF> <0033>      3
<03F0> <0034>      4
<03F2> <0036>      6
<03F5> <0039>      9
endbfchar
```

PDF association

Artifex

Even less obvious failures

Situation is worse when application writes out CMap in order that characters appear.

```
16 0 obj
<</Length 826>>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CMapType 2 def
/CMapName/R16 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
31 beginbfchar
<0003> <0003> <0020>
<0057> <0057> <0050>
<0064> <0064> <0054>
<0102> <0102> <0061>
<010f> <010f> <0062>
<0110> <0110> <0063>
<011a> <011a> <0064>
<011e> <011e> <0065>
<015a> <015a> <0068>
<0175> <0175> <006d>
<0176> <0176> <006e>
<017d><017d> <006f>
<018c><018c> <0072>
<0190> <0190> <0073>
<019a> <019a> <0074>
<019f> <019f> <fffd>
<01b5> <01b5> <0075>
<01c0> <01c0> <0076>
<01c1> <01c1> <0077>
<0357> <0357> <003a>
<0358> <0358> <002e>
<037e> <037e> <0028>
<037f> <037f> <0029>
<039b> <039b> <0040>
<03a8> <03a8> <0024>
<03ef> <03ef> <0033>
<03f0> <03f0> <0034>
<03f2> <03f2> <0036>
<03f5> <03f5> <0039>
<0439> <0439> <0025>
endbfchar
endcmap
```

Even less obvious failures

In some cases, the exact order could occur depending upon the content and the application.

```
16 0 obj
<</Length 826>>
stream
/CIDInit /ProcSet findresource begin
12 dict begin
begincmap
/CMapType 2 def
/CMapName/R16 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
```

```
31 beginbfchar
<0003> <0003> <0020>
             Space
<0057> <0057> <0050>          P
<0064> <0064> <0054>          T
<0102> <0102> <0061>          a
<010f> <010f> <0062>b
<0110> <0110> <0063>          c
<011a> <011a> <0064>          d
<011e> <011e> <0065>          e
<015a> <015a> <0068>          h
<0175> <0175> <006d>          m
<0176> <0176> <006e>          n
<017d> <017d> <006f>          o
<018c> <018c> <0072>          r
```

```
<0190> <0190> <0073>   s
<019a> <019a> <0074>   t
<019f> <019f> <fffd>
<01b5> <01b5> <0075>   u
<01c0> <01c0> <0076>   v
<01c1>  <01c1> <0077>  w
<0357> <0357> <003a>   :
<0358> <0358> <002e>   .
<037e> <037e> <0028>   (
<037f> <037f> <0029>   )
<039b> <039b> <0040>   @
<03a8> <03a8> <0024>   $
<03ef> <03ef> <0033>   3
<03f0> <03f0> <0034>   4
<03f2> <03f2> <0036>   6
<03f5> <03f5> <0039>   9
<0439> <0439> <0025>   %
endbfchar
endcmap
```

More font issues

- Previous problem was demonstrated by an Australian government study and the issue still exists today.
- Poking around quickly revealed additional issues.
- With Type 3 fonts, each character is defined by a content stream. The stream can contain path, text or image objects. The dictionary for each character is referenced in the CharProcs entry of the Type 3 font dictionary.

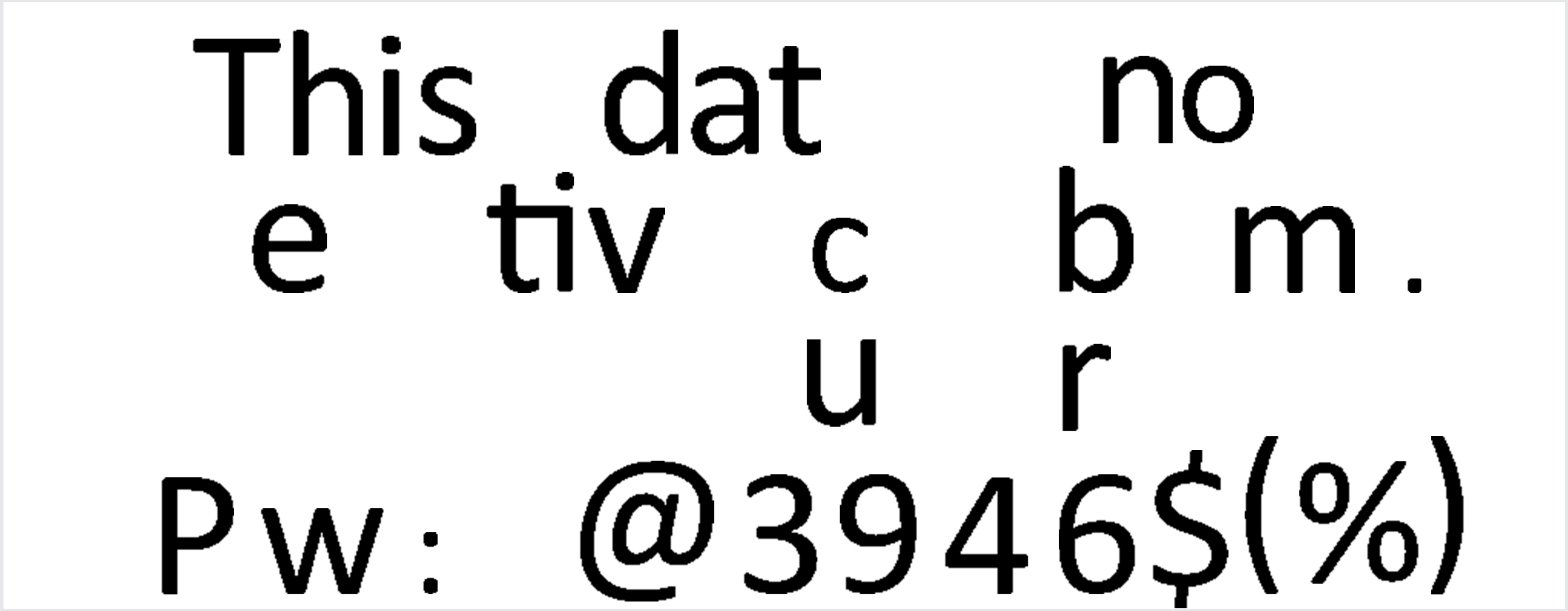
More font issues

```
55 0 obj
<</CharProcs 52 0 R /Encoding 53 0 R /FirstChar 0 /FontBBox[-1 -89 98 20] /FontMatrix[1 0 0 1 0 0] /LastChar 30
/Name/T1/Resources 54 0 R/Subtype/Type3/Type/Font
/Widths[58.4 0 63 27.5 46.9 63 57.5 40.2 63 63.2 59.8 66.8 54.2 50.8 63 95.9 30.2 63
41.9 62 85.8 32.2 107 60.8 60.8 60.8 60.8 60.8 36.4 85.8 36.4]>>
endobj
```

```
52 0 obj
<</a0 17 0 R /a1 18 0 R /a10 19 0 R /a11 20 0 R /a12 21 0 R /a13 22 0 R /a14 23 0 R /a15 24 0 R /a16 25 0 R
/a17 26 0 R /a18 27 0 R /a19 28 0 R /a2 29 0 R /a20 30 0 R /a21 31 0 R /a22 32 0 R /a23 33 0 R /a24 34 0 R
/a25 35 0 R /a26 36 0 R /a27 37 0 R /a28 38 0 R /a29 39 0 R /a3 40 0 R /a4 41 0 R /a5 42 0 R /a6 43 0 R
/a7 44 0 R /a8 45 0 R /a9 46 0 R>>
endobj
```

More font issues

- After redaction, I extracted the CharProcs streams and rendered them….



```
  This  dat         no
   e   tiv   c     b  m.
             u    r
 Pw:   @3946$(%)
```

Malicious information hiding

- Unlikely, but it is possible that information may be contained in ICC profiles (which can have private tags with information).
- Some PDF creators also place additional data in PDF objects.

Situation

- Existing solutions take a PDF file and “cut it down.”
- Risks of data ‘leaking’ through in hidden ways.
- People resort to printing the document, redacting with marker, and scanning.
- Upside: No information other than what is visible in the redacted document when it was scanned.
- Downside: Labor intensive, wasteful of paper, and subject to degrading quality.

Our approach

- Our solution, dubbed “High Security Redactions”, is essentially a digital method of the same thing.
- We build a new PDF file from a blank slate.
- User performs markups on the document using redaction annotations.
- When the document is saved, we render each page to a bitmap, and wrap bitmaps into a new PDF file.
- OCR is performed on the bitmaps and invisible text is overlaid.
- The steps that the user performs are the same using Adobe’s method, or ours.
- Our step can be used to generate a ‘high security’ version from a file that has already been redacted by Adobe’s method ensuring that information leaked from fonts does not occur.

Trade-offs

OctoberPDFest ONLINE

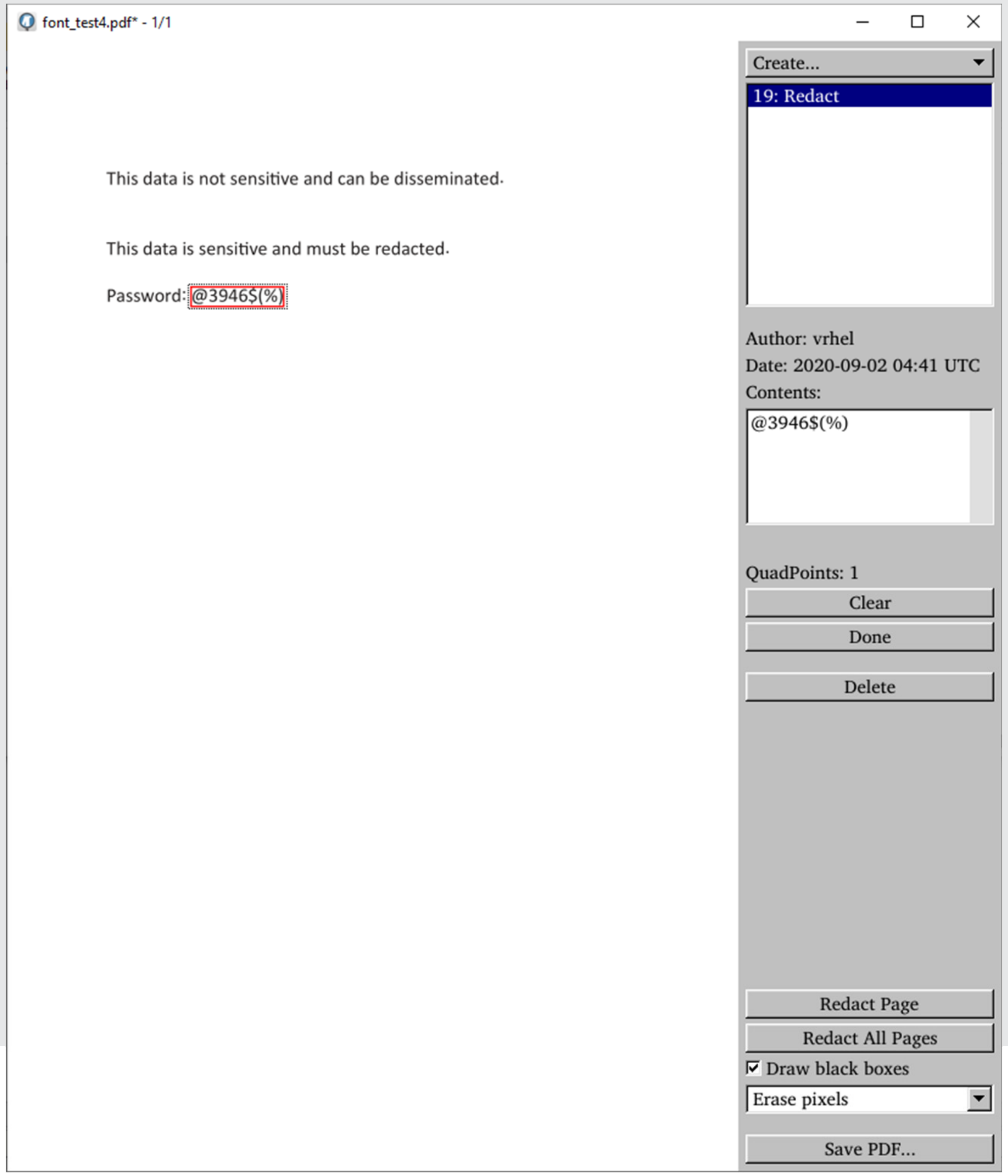
- Vector scalability loss.
- Likely larger file.
- But ensured full redaction with maintained searchability.
- No information hidden in image data, ICC profiles, fonts, or other private structures.

Open-source High Security Redaction

OctoberPDFest ONLINE

- MuPDF and Ghostscript are open-source projects that include the ability to render, manipulate, and convert PDF files.
- They have a dual commercial/AGPL license model.
- Both applications support the integration of OCR solutions. Currently the open-source Tesseract OCR solution is integrated.
- Both applications support the creation of “High Security” redaction.
- MuPDF, which is essentially a PDF toolkit with an API across multiple languages, can be used to apply redaction annotations.

Open-source High Security Redaction



font_test4.pdf* - 1/1

This data is not sensitive and can be disseminated.

This data is sensitive and must be redacted.

Password: @3946$(%)

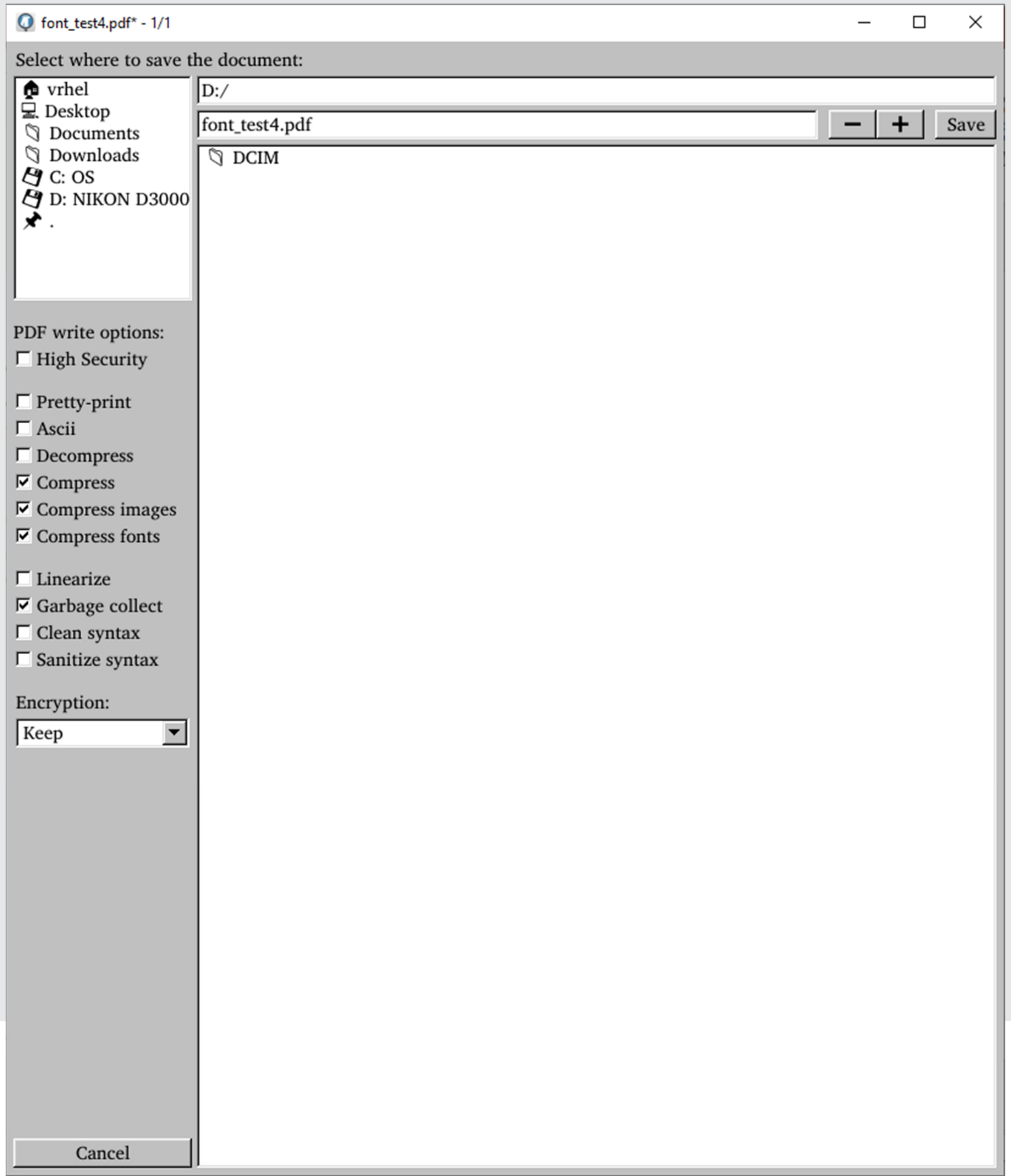
Author: vrhel
Date: 2020-09-02 04:41 UTC
Contents:
@3946$(%)

QuadPoints: 1

[✓] Draw black boxes

Erase pixels

Open-source High Security Redaction


font_test4.pdf* - 1/1

Select where to save the document:

- vrhel
- Desktop
- Documents
- Downloads
- C: OS
- D: NIKON D3000
- .

D:/

font_test4.pdf

DCIM

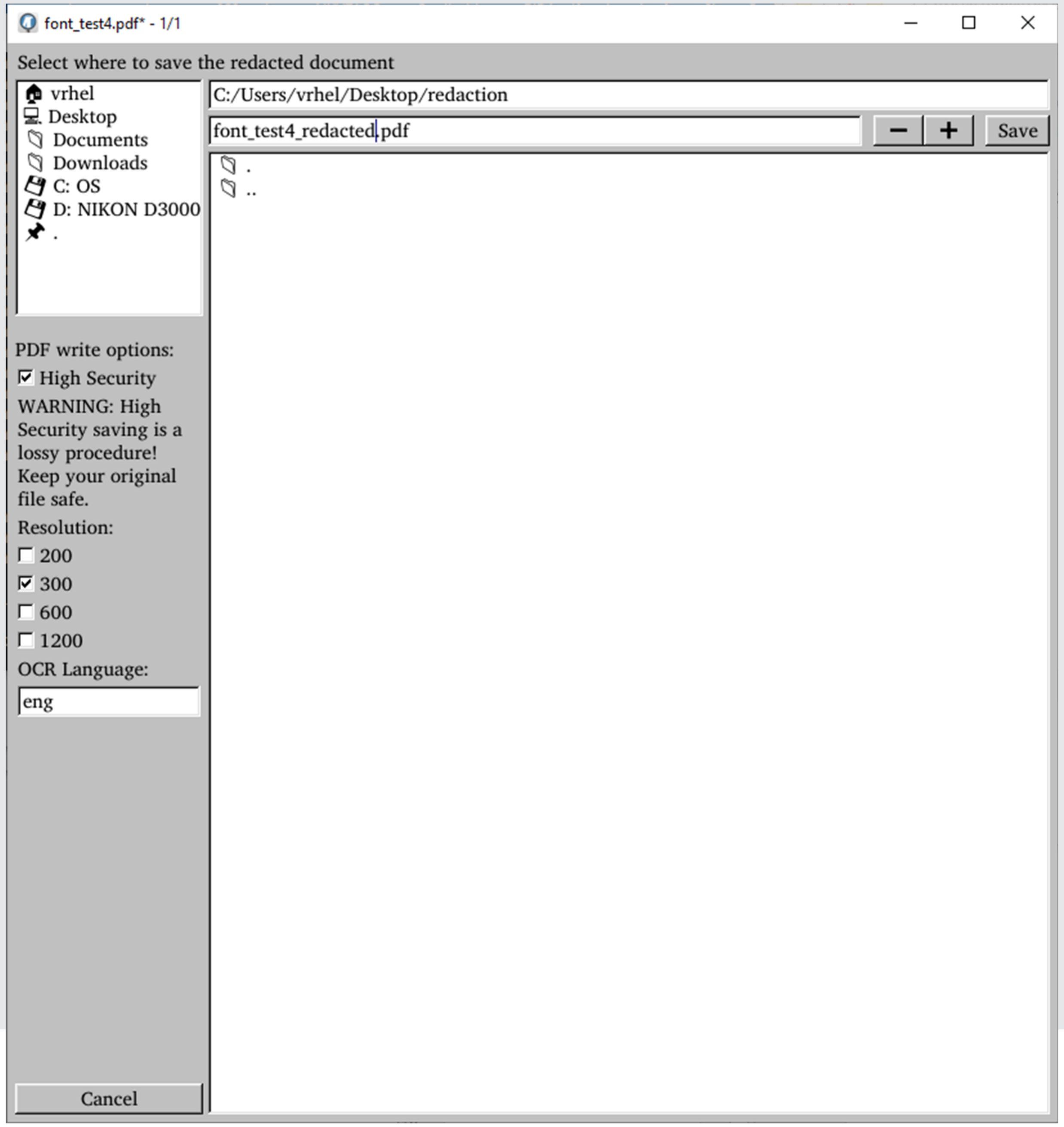
PDF write options:

- [ ] High Security
- [ ] Pretty-print
- [ ] Ascii
- [ ] Decompress
- [✓] Compress
- [✓] Compress images
- [✓] Compress fonts
- [ ] Linearize
- [✓] Garbage collect
- [ ] Clean syntax
- [ ] Sanitize syntax

Encryption: Keep

Open-source High Security Redaction

OctoberPDFest ONLINE


font_test4.pdf* - 1/1

Select where to save the redacted document

- vrhel
- Desktop
- Documents
- Downloads
- C: OS
- D: NIKON D3000
- .

C:/Users/vrhel/Desktop/redaction

font_test4_redacted.pdf

- .
- ..

PDF write options:

☑ High Security

WARNING: High Security saving is a lossy procedure! Keep your original file safe.

Resolution:

☐ 200
☑ 300
☐ 600
☐ 1200

OCR Language:

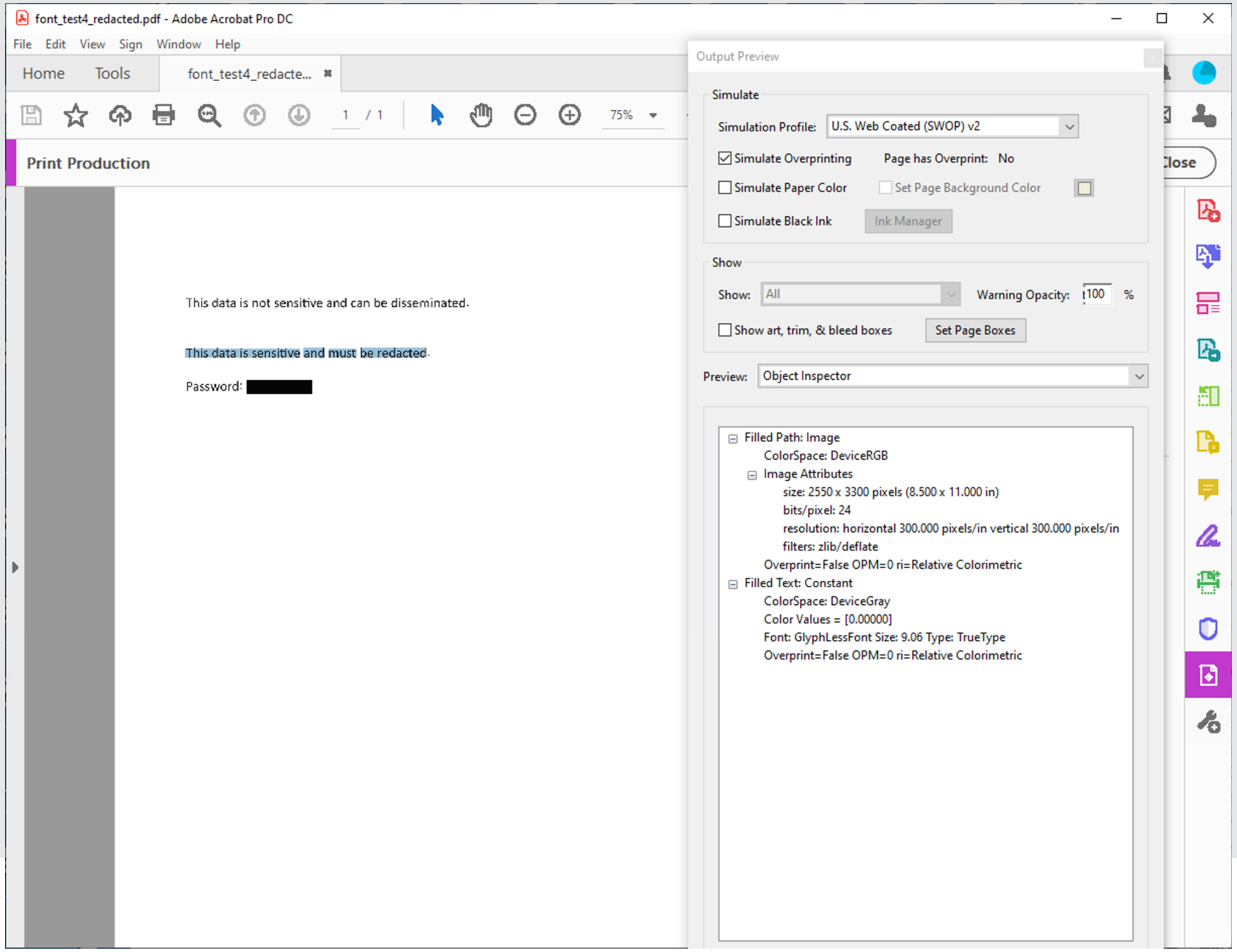
eng

PDF association

Artifex

Open-source High Security Redaction

OctoberPDFest ONLINE


font_test4_redacted.pdf - Adobe Acrobat Pro DC
File Edit View Sign Window Help
Home Tools font_test4_redacte...
1 / 1 75%
Print Production
Close

This data is not sensitive and can be disseminated.

This data is sensitive and must be redacted.

Password: [redacted]

Output Preview

Simulate

Simulation Profile: U.S. Web Coated (SWOP) v2

☑ Simulate Overprinting Page has Overprint: No

☐ Simulate Paper Color ☐ Set Page Background Color

☐ Simulate Black Ink Ink Manager

Show

Show: All Warning Opacity: 100 %

☐ Show art, trim, & bleed boxes Set Page Boxes

Preview: Object Inspector

```
⊟ Filled Path: Image
    ColorSpace: DeviceRGB
  ⊟ Image Attributes
      size: 2550 x 3300 pixels (8.500 x 11.000 in)
      bits/pixel: 24
      resolution: horizontal 300.000 pixels/in vertical 300.000 pixels/in
      filters: zlib/deflate
    Overprint=False OPM=0 ri=Relative Colorimetric
⊟ Filled Text: Constant
    ColorSpace: DeviceGray
    Color Values = [0.00000]
    Font: GlyphLessFont Size: 9.06 Type: TrueType
    Overprint=False OPM=0 ri=Relative Colorimetric
```

PDF association

Artifex

Open-source High Security Redaction

OctoberPDFest ONLINE

Using the mupdf command line tools:

```
mutool draw -o output.pdf -F ocr.pdf -c rgb -r 300 in.pdf
```

will produce a high-security redacted PDF with 300dpi full color renderings of the page.

To use ghostscript:

```
gs -o output.pdf -sDEVICE=pdfocr24 -r300 in.pdf
```

will do the same.

30 years of technology innovation

OctoberPDFest ONLINE

1990	• 1988 Ghostscript development began • 1990 first release of Ghostscript PS interpreter • 1993 Artifex released first PDF interpreter before everyone except Adobe • 1996 Artifex released first PCLXL interpreter before anyone but HP • 1998 Artifex released PCL6 (PCL5/PCLXL) interpreter
2000	• 2000 Artifex released PostScript 3 interpreter • 2002 Artifex released first PDF 1.4 transparency interpreter before anyone but Adobe • 2002 Kyocera released first native PDF printer using GhostPDF (now over 20 products and millions of units sold) • 2003 MuPDF development began • 2006 Artifex released PDF 1.7 interpreter • 2008 Artifex released XPS interpreter
2010	• 2010 Artifex released new color-handling • 2012 MuPDF 1.0 was released • 2013 MuPDF added JavaScript and all interactive features • 2014 Artifex acquired SmartOffice with MS Office compatibility • 2014 Adobe licensed MuPDF for Adobe Digital Publishing Suite Android mobile • 2015 Google licensed Ghostscript as cloud solution • 2018 Cobalt Acceleration Library development began • 2019 MuPDF adds WASM build targets • 2020 MuPDF adds Python bindings • 2020 Artifex engineers develop ”High Security Redactions”

PDF association

Artifex

About Artifex

Artifex is committed to providing quality software solutions.

- Roots in open-source with Ghostscript and MuPDF
- Partners with over 150 OEM leaders such as Adobe, Google, Blackberry, Garmin, Dropbox
- Talented engineering staff in U.S., Europe, and Asia
- Over 100 consecutive quarters of profitability
- Deeply committed to product quality and service
- Dedicated to providing prompt and professional technical support and consulting services to our customers
- Flexible licensing arrangements based on individual use case

Global Service & Support

The Artifex Advantage

OctoberPDFest ONLINE

- We build long-term relationships based on strong technical knowledge, outstanding service and a willingness to invest in our partner relationships.
- Over 150 OEM partners representing some of the biggest names in technology depend on Artifex software solutions.

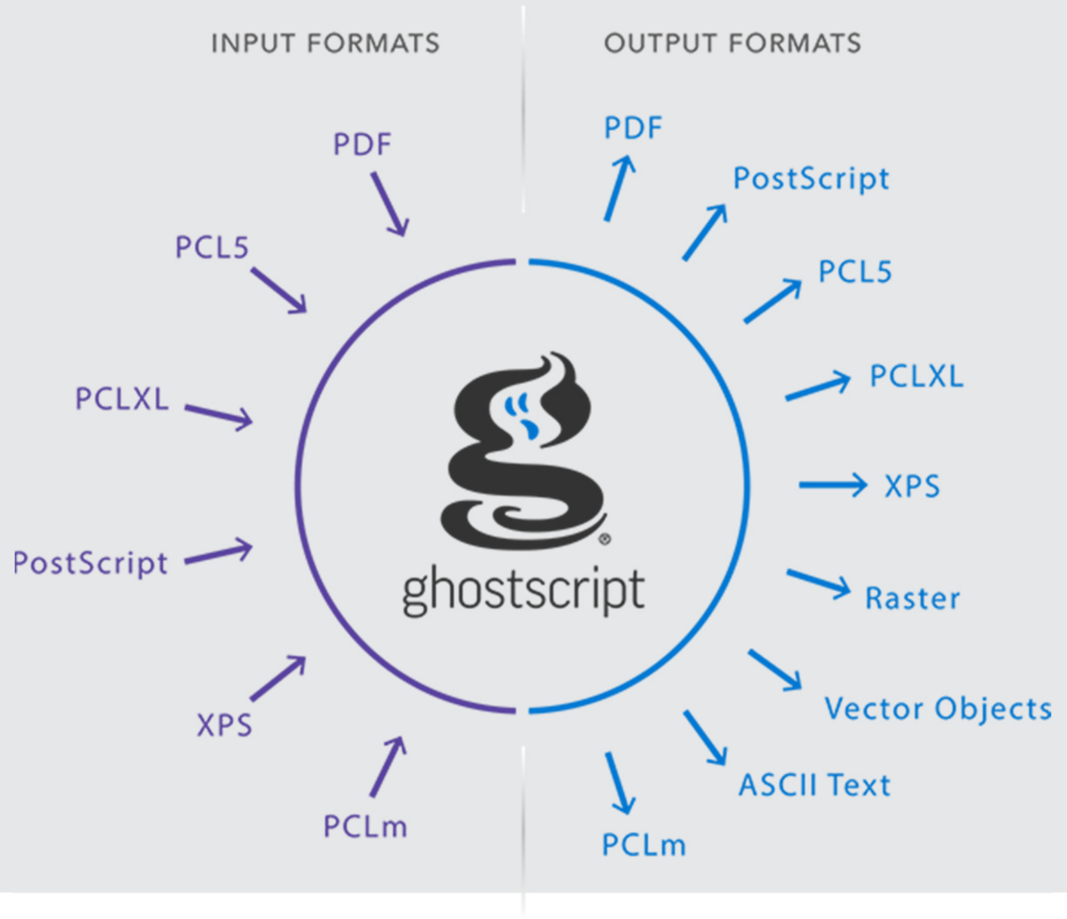
Dropbox Adobe Google ActivePDF® opentext™
GERBER TECHNOLOGY BlackBerry Nasdaq Boston Scientific
KYOCERA RICOH hp KIP ergosoft rip solutions
IBM océ intuit SIEMENS
XEROX pitney bowes brother at your side CALDERA

Ghostscript

Ghostscript has the most comprehensive conversion capabilities and flexibility of any software in its class.

- PDF 1.2 → 2.0
- Mopria, AirPrint
- PostScript Level 1 → 3
- PCL5e, PCL5c, PCLXL 3.0
- HP-GL, HP-GL/2, HP RTL
- XPS
- Tested using Quality Logic FTS, ATS, CET suites as well as proprietary, customer and user files

The #1 PDL Conversion Tool Available Today

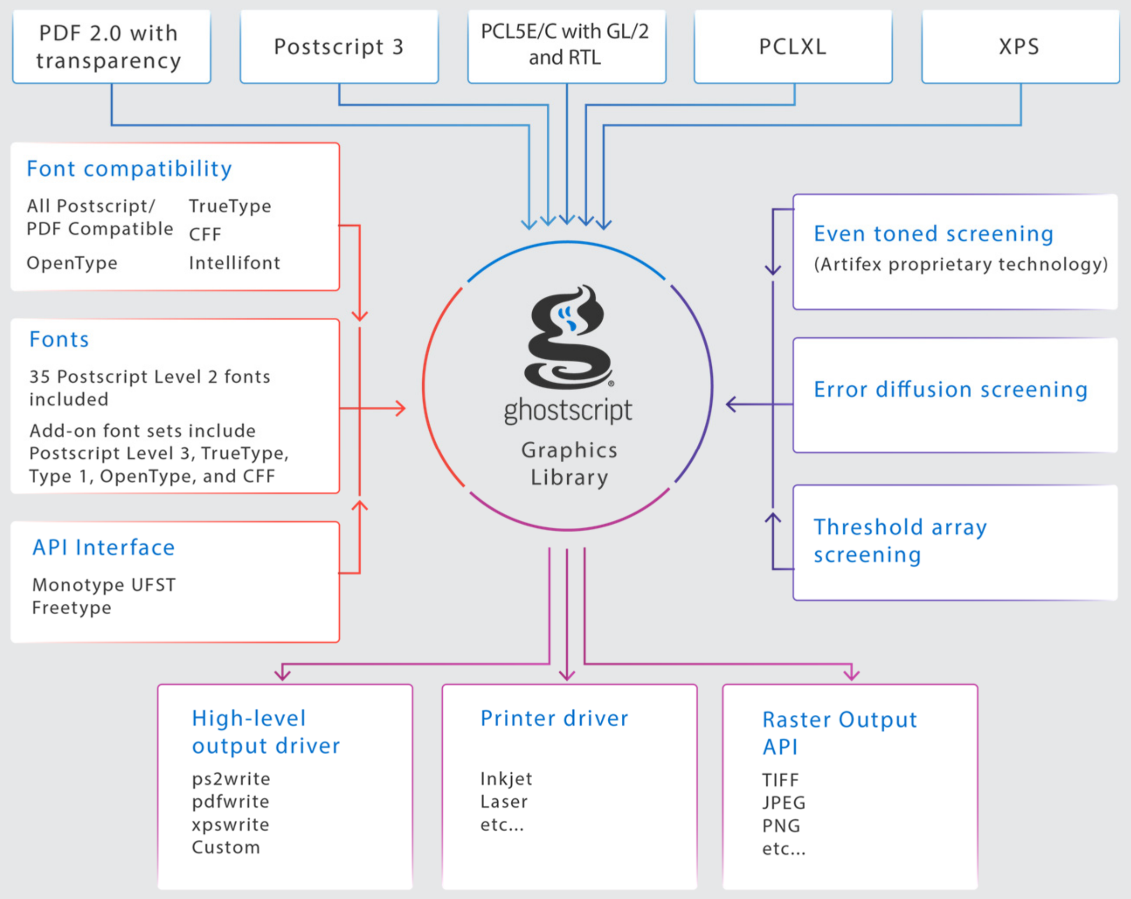


Ghostscript

The modular graphics library with its proprietary color management system and advanced screening technology make Ghostscript a flexible and robust solution.

- Comprehensive font solution
- Fully compatible interpreters
- Multi-platform support (Unix, Linux, MS Windows, macOS X, VxWorks)
- Graphic object dependent color management technology
- High-performance, lightweight solution (3MB)
- Vast range of customizable features

30 Years of Technology Evolution

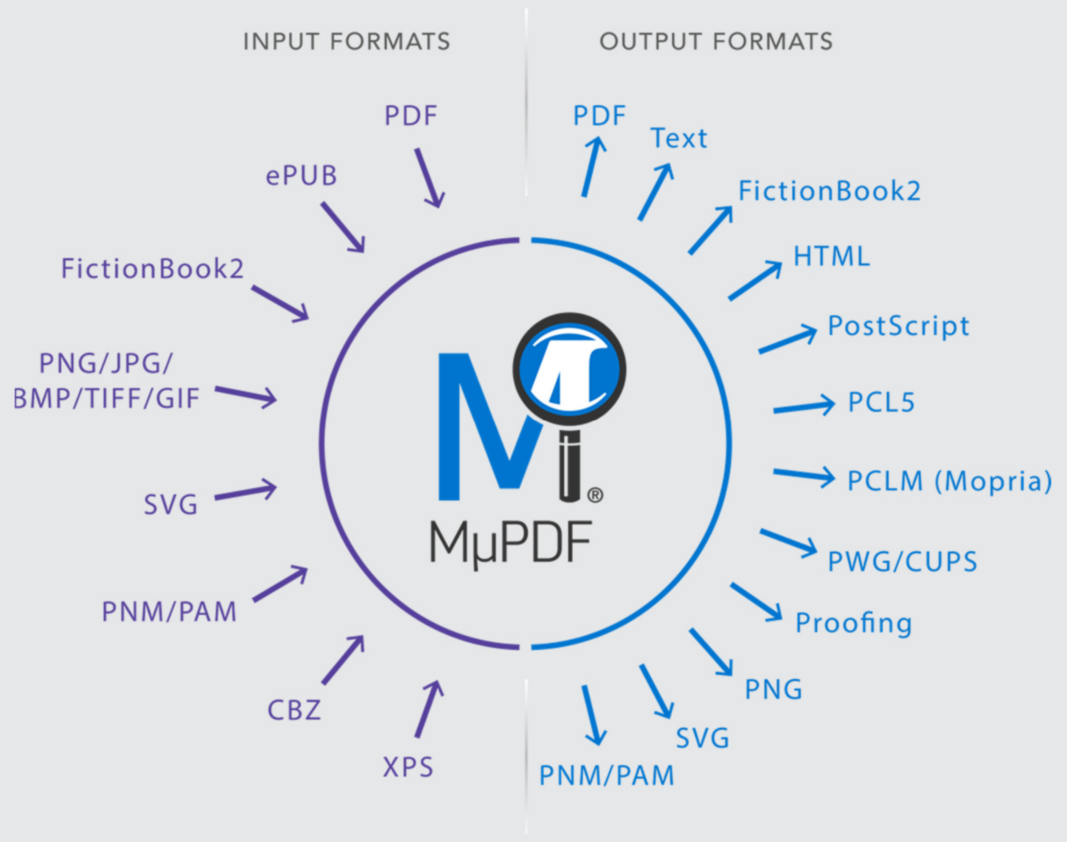

PDF 2.0 with transparency
Postscript 3
PCL5E/C with GL/2 and RTL
PCLXL
XPS
Font compatibility
All Postscript/ PDF Compatible
TrueType
CFF
OpenType
Intellifont
Fonts
35 Postscript Level 2 fonts included
Add-on font sets include Postscript Level 3, TrueType, Type 1, OpenType, and CFF
API Interface
Monotype UFST
Freetype
ghostscript
Graphics Library
Even toned screening
(Artifex proprietary technology)
Error diffusion screening
Threshold array screening
High-level output driver
ps2write
pdfwrite
xpswrite
Custom
Printer driver
Inkjet
Laser
etc...
Raster Output API
TIFF
JPEG
PNG
etc...

MuPDF

A highly versatile, customizable PDF and XPS interpreter, viewer or toolkit solution.

- Integrates with a wide variety of embedded, host-based, mobile or cloud applications
- High-fidelity, accurate PDF rendering
- Interactive PDF features (annotations, form filling, etc.)
- Small footprint (2MB)
- Multi-platform support (Android, iOS, Win, macOS, Linux, Unix)

Ultra-Small, Ultra-Fast PDF Rendering Solution



SmartOffice

View, edit, create, print, present and share Microsoft Office and PDF documents via your mobile device.

- SmartOffice handles all major document formats on mobile or embedded devices:
  - MS Word, Excel, PowerPoint, PDF, image files (.jpg, .png, .gif), text files (.txt), HWP
  - iOS, Android, Windows, macOS platforms
- Convert MS docs to PDF
- Interactive PDF features
- Securely integrates into your enterprise document workflow
- Small download size ensures fast loading and high-quality performance

Made for Mobile Document Productivity